

AD-A014 843

SAINT: VOLUME I. SYSTEMS ANALYSIS
OF INTEGRATED NETWORK OF TASKS

A. Alan B. Pritsker, et al

Purdue University

Prepared for:

Aerospace Medical Research Laboratory

April 1974

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE

268254

ADA014843

AMRL-TR-73-126



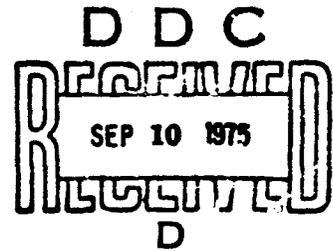
**SAINT:
VOLUME I. SYSTEMS ANALYSIS OF INTEGRATED
NETWORK OF TASKS**

**A. ALAN B. PRITSKER
DAVID B. WORTMAN
CHARLES S. SPUM
PURDUE UNIVERSITY**

**GERALD P. CHUBB
DEBORAH J. SEIFERT**

AEROSPACE MEDICAL RESEARCH LABORATORY

APRIL 1974



Approved for public release; distribution unlimited

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
US Department of Commerce
Springfield, VA. 22151

**AEROSPACE MEDICAL RESEARCH LABORATORY
AEROSPACE MEDICAL DIVISION
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433**

ADDRESS BY	
USE	With Index <input type="checkbox"/>
DO	Self Index <input type="checkbox"/>
REPRODUCTION	<input type="checkbox"/>
REPRODUCTION	<input type="checkbox"/>
BY	
SUBSTITUTE/STAMP WITH INDEX	
MAIL	AIRMAIL
A	

NOTICES

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Organizations and individuals receiving announcements or reports via the Aerospace Medical Research Laboratory automatic mailing lists should submit the addressograph plate stamp on the report envelope or refer to the code number when corresponding about change of address or cancellation.

Do not return this copy. Retain or destroy.

Please do not request copies of this report from Aerospace Medical Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
 5285 Port Royal Road
 Springfield, Virginia 22151

This report has been reviewed and cleared for open publication and/or public release by the appropriate Office of Information (OI) in accordance with AFR 190-17 and DODD 5230.0. There is no objection to unlimited distribution of this report to the public at large, or by DDC to the National Technical Information Service (NTIS).

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Julien M. Christensen
 JULIEN M. CHRISTENSEN, Ph.D.
 Director, Human Engineering Division
 6570 Aerospace Medical Research Laboratory

1 X

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AMRL-TR-73-126	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SAINT: Volume I. Systems Analysis of Integrated Networks of Tasks		5. TYPE OF REPORT & PERIOD COVERED Final Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) A. Alan B. Pritsker* Gerald P. Chubb** David B. Wortman* Deborah J. Seifert** Charles S. Seum*		8. CONTRACT OR GRANT NUMBER(s) F33615-73-C-4029
9. PERFORMING ORGANIZATION NAME AND ADDRESS School of Industrial Engineering* Aerospace Medical Purdue University Research Lab.** West Lafayette, Indiana 47907 WPAFB OH 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62202F; 7184; 718409; 71840934
11. CONTROLLING OFFICE NAME AND ADDRESS Aerospace Medical Research Laboratory Aerospace Medical Division, AFSC Wright-Patterson Air Force Base, Ohio 45433		12. REPORT DATE April 1974
		13. NUMBER OF PAGES 61
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) DDC RECEIVED SEP 10 1975 REGISTRY D		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Computers Mission Analysis Networks Modeling Vulnerability/Survivability Simulation Operator Loading Man-Machine Relationships Operations Research Crew Performance		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A combined network modeling and simulation technique called SAINT is developed for analyzing a set of problems from the field of human engineering. SAINT is designed to model in network form sets of tasks performed during the course of a mission. SAINT obtains mission performance measures for networks which represent a mission consisting of a set of tasks performed by a crew of operators having a complement of equipment in the face of environmental factors.		

A simulation approach is used to obtain the performance measures. Human Engineering considerations are included through parameters associated with tasks, precedence relations between tasks and factors affecting crew performance. SAINT is both a modeling procedure and an analysis technique. The analysis is performed by a digital computer program which accepts input data concerning tasks and performs two types of analysis. First, SAINT performs benchmark iterations to obtain estimates of the time requirements on the operators performing the mission. Estimates of essential time remaining, nonessential time remaining, and waiting time for various segments of a mission are computed. Second, SAINT obtains mission performance measures which provide estimates of the probability of successfully completing the mission under stress and adverse environmental conditions. This report presents the methods for using SAINT, including the preparation of input data and the interpretation of output reports.

SUMMARY

This first of two volumes describes the technical details of a new technique for human engineering: SAINT (Systems Analysis of Integrated Networks of Tasks). The second volume (AMRL-TR-73-128) is a user's manual for the computer program which analyzes such networks.

SAINT provides a graphic symbol set for diagramming event sequences. The network symbols developed for P-GERT (Precedence-GERT, GERT being a Graphical Evaluation and Review Technique) were used as a basis for SAINT. GERT was itself developed previously to aid engineers interested in applying network theory and simulation to operations and systems analysis problems. SAINT extends these capabilities to allow a description of human activities in terms of a set of tasks performed by a crew or set of operators.

The computer program for analyzing task/activity networks includes all the human performance dynamics proposed by Siegel and Wolf in their two-man operator simulation model. However, because of the enhanced capabilities of the SAINT symbol set, many of the model constructs used by Siegel and Wolf have been generalized for crew sizes of up to eleven operators. Although not discussed in detail, the computer routines also include the modifications made to the Siegel-Wolf model to permit a consideration of the impact nuclear weapons effects could have on human performance. Consequently, the SAINT routines can be used as an aid to determining the vulnerability/survivability of manned systems.

By design, the SAINT technique does not require the user to perform any computer programming. Users are assumed to be knowledgeable of task analysis or methods engineering. The results of a task analysis are used as the inputs to the SAINT computer program. The output of SAINT consists of task and mission performance estimates.

The SAINT computer programs are coded in FORTRAN IV and should be executable on almost any digital computer having a FORTRAN compiler.

While the technique has been developed to examine human factors affecting the outcomes of military missions, the technique is generally applicable to non-military activities and should, therefore, prove useful in industrial, transportation and consumer products studies of interest to human factors specialists.

FOREWORD

This study was initiated by the Human Engineering Division, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio 45433. The research was conducted by Purdue University, West Lafayette, Indiana 47907. A. Alan B. Pritsker was the principal investigator for Purdue University. The work was performed in support of project 7184 "Human Engineering for Air Force Systems," task 718409 "Man-Machine Systems Effectiveness." This effort is part of the support provided in response to AFSWC-TN-69-1 "Systems Approach to the Nuclear Vulnerability Assessment of Man in Air Force Systems," with funding provided from project 8809, "Nuclear Vulnerability and Hardening Technology," task 880903, "Analysis of S/V of Air Force Aeronautical Systems in a Nuclear Environment." The research sponsored by this contract was performed between September 1972 and August 1973, under Air Force contract F33615-73-C-4029.

Dr. Arthur D. Siegel of Applied Psychological Services, Inc., Wayne, Pennsylvania 19087 is to be thanked for his assistance and cooperation in this work which extends the pioneering effort he and his associates contributed under contracts F33615-70-C-1798 and F33615-69-C-1880 as well as the Navy-funded development of the original, two-man, operator simulation model.

TABLE OF CONTENTS

	<u>Page</u>
SECTION I - INTRODUCTION	1
Rationale and Background	2
Evolution	2
Design Philosophy	4
SECTION II - TASK-ORIENTED CONCEPTS	6
Task Input	6
Task Duration	7
Task Essentiality	9
Task Type	9
Task Class	12
Task Output	12
Additional Task-oriented Concepts	14
Statistical Information Collection	15
SECTION III - OPERATOR-ORIENTED CONCEPTS	17
Speed	17
Accuracy	19
Stress	20
Goal Gradients	33
Cohesiveness	34
Computational Sequence	34
SECTION IV - SAINT MODELING	36
Branch Symbolism	39
Operator Symbolism	40
Examples	42
SECTION V - SUMMARY	49
REFERENCES	51

LIST OF FIGURES

<u>Figure Number</u>		<u>Page</u>
1	Probability of Success Modification for Different Task Accuracy Factors	21
2	Probability of Skipping a Task as a Function of Stress and Essentiality	32
3	Task Symbol and Terminology	37
4	Branching Types, Symbols and Terminology ..	41
5	A Series of Tasks by a Single Operator	44
6	Parallel Tasks Leading to a Joint Task Involving Two Operators	44
7	A Joint Task Followed by Two Parallel Tasks Involving Two Operators	46
8	Parallel Tasks for Two Operators Sepa- rated by an Either Task	46
9	Network Modification Based on Task Completion Sequence	47
10	Clearing of Operators	47

LIST OF TABLES

<u>Table Number</u>		<u>Page</u>
1	Z Values as a Function of Stress and Essentiality	30
2	Goal Gradient Additive Factor	33

SECTION I

INTRODUCTION

Human engineering, as an applied psychology of man-machine relationships has attempted to apply the methods of experimental psychology to the study of human behavior in a systems context. The product of such endeavors has been the specification of design standards, the generation of data for design tradeoff studies, or specific recommendations concerning proposed design alternatives. However, the achievement of the implied objective - the support of design and systems engineers - has often been frustrated by the difficulty encountered in translating human performance into systems outcomes. It matters little that one can design equipment or procedures so they save time or curtail human errors if these benefits do not somehow manifest themselves in terms of mission relevant consequences: increased system effectiveness and/or reduced cost.

One technique for accomplishing such determinations is simulation. With simulation, the events and activities associated with man-machine performance are portrayed in the context of the application of interest. The systems' performance in a mission is represented by the model's portrayal of man-machine activities in response to mission events.

The ability of simulation to evaluate hypothetical mission performance in some satisfactory sense has led to its wide-spread use. Simulation has been used only on a limited basis for human engineering design assessment, and the common practice has been to assume that because each system is unique, the model for and simulation of that system must also be unique. Few basic modeling concepts have been developed for use in simulating manned systems. The carryover benefits from simulation-to-simulation have been small except for the experience gained by those who designed and ran the studies.

The goal of the current research is to develop basic simulation concepts that would permit the transferral of knowledge from one project to another. It is not the intent of the research to provide a tool or set of concepts that will be usable for a large class of human engineering problems. The objective is to provide a tool that permits more realistic mission simulation of manned systems.

The set of problems considered focuses on task allocation, operator workload, and environmental stressors. It is conjectured that network concepts and symbols can be developed that will permit the modeling of one or more operators performing an assigned set of tasks within the context of a specific

mission and the operating environment for a mission. Once the network concepts and symbols are designed, a simulation program can be developed for analyzing mission performance as a function of operator and environmental variables. By definition, a mission is modeled as a network of tasks. Operators perform the tasks and in so doing accomplish the objective of the mission. A mission is completed when a specified end task or tasks have been completed. Mission performance is related to which tasks are achieved, the manner in which they are achieved, and the times at which they are completed.

Rationale and Background

Communications in research play a fundamental role especially when interdisciplinary activities are involved. Mental images and concepts are satisfactory as long as a single researcher is working on a problem. As soon as two or more individuals are working together, a vehicle for expressing one's ideas and concepts is necessary. The use of networks or graphs as communication vehicles for researchers is well established. Examples of networks are circuit diagrams, free body diagrams, signal flow graphs, block diagrams and PERT networks. Networks are models of systems. These models may be used for communication purposes and/or analysis purposes. In many cases it is the former purpose which is significant as it permits a concise, explicit definition of the pertinent concepts the researcher wishes to convey. Once a network is prescribed, effort can be concentrated on analyzing the system by analyzing the network model. When the network model can be used for both descriptive and analysis procedures, the researcher has a significant tool at his disposal.

The human factors specialist has long advocated the use of operational sequence diagrams (4)*, function flow logic block diagrams (16) and to some extent, models (5). Besides Siegel and Wolf's early use of Monte Carlo techniques to simulate operators performing discrete tasks, there are precedents for using Monte Carlo simulation to portray manual tracking tasks as well (1). What appears to be required is an integrating framework that not only assimilates and consolidates these previous achievements but allows for the systematic enrichment of such technical tools.

Evolution

Although there are many projects that have contributed to the research reported herein, there were two parallel developments that led directly to the creation and implementation

* Numbers in parentheses refer to reference section.

of SAINT. A two-man model of operators performing a specified task sequence has been developed by Siegel and Wolf. The conceptual nature of this model and the history of its development are well documented (11). A flow diagram of the FORTRAN IV program and a discussion of possible extensions to the model have also been reported (12). Modifications made to treat the impact ionizing radiation has on performance are reported by Chubb (3).

The significance of the work of Siegel and his associates is the concept of modeling a task in terms of the task type, operator characteristics, environmental characteristics, and a mission scenario. Each task is modeled as a discrete event, consuming time and terminating as a success or failure. A computer program was developed for simulating the set of tasks in a mission to assess the consequences of operator errors, skill proficiency and other factors on mission performance. Through this procedure, one could examine the impact specific characteristics of tasks, operators, and the environment might have on a mission. Consideration is given to the time available for the mission, the time taken to perform a task as a function of stress, the probability of successfully completing a task as a function of stress, operator proficiency (in terms of speed and accuracy), and the impact of such mission related stressors such as exposure to ionizing radiation. Seifert and Chubb (10) review a modified version of the Siegel-Wolf model, illustrating its use for vulnerability/survivability assessments.

The second major development which contributed directly to the evolution of SAINT is the Graphical Evaluation and Review Technique, GERT. GERT is a generalized network technique for analyzing a network consisting of a set of activities. Here activities are tasks, but GERT does not require that activities be tasks. During the 1960's and early '70's, the development of GERT concentrated on network structure and generalized procedures for describing activities. These developments by Pritsker are reported in numerous papers and technical reports (6, 7, 8), and more recently are documented in a book by Whithouse (14). Both analytic and simulation techniques were developed for analyzing GERT networks.

The research and development on GERT concentrated on the development of graphical symbols which would permit the modeling of diverse systems in network form. Thus, generalized branching procedures, randomly distributed activity durations, network modification mechanisms, logical requirements for starting activities, and statistical data collection methods were developed as integral parts of GERT.

The differences between GERT and the Siegel-Wolf model are both conceptual and mechanistic. Whereas GERT has complex and sophisticated network symbols and constructs, it has a very aggregate method for describing the performance of activities. On the contrary, the Siegel-Wolf model has a complex and sophisticated procedure for prescribing task performance, but has a simple set of network symbols and constructs. Further, while the GERT programs are written in FORTRAN IV, they make extensive use of specialized and efficient programming techniques whereas the Siegel-Wolf model is coded in a straightforward manner in FORTRAN IV. SAINT was designed to capture the best features of the Siegel-Wolf and the GERT models. The version of the Siegel-Wolf model selected for incorporation into SAINT is the 1971 version (12). An activity-on-node version of GERT called Precedence GERT (7) was selected for incorporation into SAINT.

Design Philosophy

The basic philosophy in designing SAINT is that additions, modifications and deletions are to be expected. Because of this, the SAINT computer program was designed in a modular form to facilitate adaptation. The computer program is based on GASP IIA (9) which is well documented and provides the necessary support programs for a simulation which is modular and yet efficient. GASP IIA and, hence, SAINT is FORTRAN based and can be run on most computers having a FORTRAN compiler.

A next-event simulation philosophy has been adopted. All changes in the status of the system occur when a task is completed. During the time interval from one task completion to the next task completion the status of the system remains unchanged, i.e., idle operators remain idle, operators working on specified tasks continue to work on these tasks, etc. It is only when a task is completed that other tasks can be started, and operators can be assigned to tasks. The word task is used here in a generic sense and can include both physical tasks or other events. An example of the latter is a programmed delay or a "clock" task that interrupts the performance of other operator tasks. In this case, the status change occurs at the end of the clock task which can then affect the status of other tasks while they are in progress.

To simplify the presentation in this report, the concepts relating to tasks and operators have been separated and are presented in Sections II and III, respectively. In Section IV, the SAINT modeling procedures and symbol set are presented. Section V then integrates the information presented in Sections II, III and IV. The end of Section V includes a summary and recommendations for future work.

The detailed information for using SAINT is presented in a companion report, Volume II (AMRL-TR-73-128) entitled: "SAINT: User's Manual" (15), which includes definitions and a listing of the FORTRAN IV routines. The SAINT computer program has been implemented on Purdue's CDC 6500 and the IBM 360 computer facility at the Aerospace Medical Research Laboratory. The Aerospace Medical Research Laboratory may be contacted for additional information and assistance in implementing or using these routines for applications of interest to the reader.

SECTION II

TASK-ORIENTED CONCEPTS

As drawn from the Siegel-Wolf model, the basic element of a SAINT network is a task. In SAINT, tasks are related to one another by precedence relations. A precedence relation stipulates that a task can be initiated only when some preceding event, task or other condition has occurred. In SAINT networks, tasks are represented by the nodes of the network while the precedence relations are represented by branches between tasks. The precedence relations indicate the flow of operators through the network as well as indicating which tasks can be started based on the completion of a particular task. Tasks can, when completed, modify the network by the addition, removal, or replacement of precedence relations, tasks, or task descriptors (e.g. the time distribution or its parameter values). Thus, a SAINT network consists of tasks which when completed can cause precedence requirements for other tasks to be satisfied. At the same time SAINT provides a mechanism for modifying the network due to the sequence in which tasks are performed, conditions obtained when a task is executed or by contingency events that affect what the operators are obligated to do.

A SAINT task has associated with it an input side, a task description, and an output side. The input side of a task specifies the number and nature of predecessor tasks that must be completed before the task can be started or "released." The task description consists of a number of parameters associated with task performance and the specification of the kinds of statistical information to be collected. The output side represents branching to be performed upon completion of the task. This branching can reflect the natural sequence of events or it can portray the outcomes of decisions made by one or more operators.

Task Input

The input side of a SAINT task specifies the number of predecessor tasks that must be completed before the task can be started. It also specifies whether the required number of predecessor completions must all be different. The conceptual discussion which follows is presented in the abstract and may be somewhat difficult to conceptualize fully with one reading. It is suggested that this material be read again after the examples in Section IV have been reviewed and the reader is more familiar with these concepts.

Each time a predecessor task is completed and the branch representing the precedence relation is taken, a requirement for

starting the task is satisfied. The task is released* for starting when a specified number of requirements is satisfied. Thus, associated with the input side of a task is the number of requirements to release the task. Since tasks can be released more than once (i.e., feedback is permitted), and since the first time usually represents a special case, the design of SAINT allows for two values to be associated with the number of requirements to release a task: 1) the number of requirements to release the task for the first time; and 2) the number of requirements to release the task after the first time.

For the input side of a task, a specification is also made as to whether all requirements for the task must be different (as opposed to allowing repeated occurrences of a single predecessor task) in order for the task to be released. If this is not specified, then sequential completions of some single predecessor task can be used to satisfy the number of requirements for a task.

By combining the two input specifications, various logic operations can be modeled. The logic operation of "AND" is modeled when the number of requirements must equal the number of predecessor tasks and all of these requirements must be different tasks. An OR logic operation specifies that only one predecessor task completion out of several tasks is required to release the task in question. A "Majority Voting" logic is an operation which specifies that more than half of the incoming branches to a task must be satisfied and that they must represent different task completions. As can be seen, the SAINT model specifications permit a large degree of flexibility in specifying precedence requirements.

Task Duration

Task duration or the time to perform a task is characterized by both a distribution type and a parameter set. Sampling can be performed from any of the following distribution types:

1. Constant
2. Normal
3. Uniform
4. Erlang

* The term "released" is used instead of "started" because all predecessor tasks can be completed but the task not started due to a resource conflict, i.e., two or more tasks wanting to use the same resource at the same time.

5. Lognormal
6. Poisson
7. Beta
8. Gamma
9. Beta fitted to three estimates
10. Constant divided by a scale factor
11. Triangular

In SAINT, samples are obtained from these distributions by using whatever information the user provides in a parameter set identified through a parameter set number. The method for doing so is not discussed here but is treated in Volume II. The parameters in a set detail such information as the mean, standard deviation, minimum value, and maximum value associated with the distribution. The samples are obtained such that if a sample is less than the minimum value, the sample value is given the minimum value. Similarly, if the sample value is greater than the maximum value, the sample value is assigned the maximum value. Strictly speaking, this technique does not provide sampling from a truncated distribution, but rather sampling from a distribution with a given probability of obtaining the minimum and maximum values.

Task duration is not solely dictated by the sample drawn but can be affected by operator characteristics or by environmental factors. Currently, only the degradation imposed by exposure to a nuclear weapons environment has been considered, but other factors can be added in future studies. Operator characteristics that have an effect on task duration currently include the operator's speed and the "time stress" to which this operator is subject at the time the task is executed. The method by which these characteristics affect task duration is discussed in the next section.

Task duration may also be modified by a task adjustment factor. This adjustment factor is used to reflect an increase or decrease in the time to perform a task if it is repeated. Suppose a task takes 10 time units to be performed and has an adjustment factor of 0.90. The first time the task is performed, its duration will be 10 time units. The second time it is performed it will take 10×0.90 or 9 time units. The third time it will take $10 \times 0.90 \times 0.90$ or 8.1 time units. If the same task had an adjustment factor of 1.10, it would take 10×1.10 or 11 time units to perform the task for the second time and $10 \times 1.10 \times 1.10$ or 12.1 time units the third time. Thus, the task adjustment factor provides a basis for a continual modification of task duration.

If a task, or operator performing a task, is subject to radiation exposure, or any other performance degrading stressor, the task duration is also subject to modification.

A complete discussion of the rationale, approach and method for incorporating the effects of exposure to the ionizing radiation of a nuclear weapon has been presented previously (3), and will not be included in this report.

Task Essentiality

Each task performed by one or more operators has associated with it a graded essentiality factor (12). The effect of task essentiality is to allow, under certain circumstances, less-essential tasks to be skipped more frequently in the interest of conserving time to reduce time stress. Task essentiality plays a role in the simulation only when the time available to complete the remaining portion of the mission becomes critical. When time remaining is not of a critical nature, all tasks are performed. However, when there is insufficient time available after the release of an operator-constrained task, a decision is made whether to perform the next task or to omit it. This decision is made in accordance with task essentiality. In essence, the essentiality value assigned to a task by the user is a measure of how important the user believes a particular task is to overall mission success.

Currently, task essentiality may be graded on a scale from zero to one as suggested by Siegel et al. (12). A task with an essentiality of zero will be performed only when the amount of essential time remaining is less than the remaining time available.*

However, if the essential time remaining is greater than the time available, the task may be skipped. The probability of performing a task in this situation is related to the essentiality assigned to it as will be discussed in the next section. A task with an essentiality of one will always be performed. Thus, an essentiality of one indicates a highly essential task.

Task Type

Each task in a SAINT network is characterized by a task type. There are six possible task types in SAINT:

* Essential time remaining is an estimate of the average time that would be required to perform the tasks in the remainder of the mission that are considered to be essential. Remaining time available is the difference between the prescribed mission completion time and the current time.

- S. Single operator task
- J. Joint task
- Q. Equipment task
- C. Cyclic task
- E. Either task
- F. Gap filler task

The last two are additions to the task types recognized in the Siegel-Wolf model. The task type chosen affects the conditions under which a task is performed.

A "single-operator task" is performed by only one operator. Only one operator may be specified as being associated with this type of task. The task is not performed until two conditions are met. First, the precedence requirements must be satisfied. Second, the required operator must be available to perform the task. In the current representation, an operator can not perform more than one task at a time.

It should be mentioned, however, that an individual need not be represented by a single operator. It is possible to portray task activities of an individual in terms of several operators. This allows the modeler to represent the operations of vision, right and left hands, right and left feet, etc. as separate operators. This obviously requires a more complex network model, additional data, and considerable attention to detail. The desirability of considering such details is left to the discretion of the user and must be evaluated in light of the problem to be solved, the resources available, and other constraints.

The important point here is simply that the SAINT concepts can be adapted to different levels of description, and the reader is cautioned against inferring that restrictions exist simply because of the terminology used to describe a particular concept. Once the full import of the concept is properly perceived, it is often apparent that other analogies can be proposed that permit one to associate different terms with a concept or network symbol. One's ability to develop such analogies improves as experience is gained in using the symbol set and sharing such experience with other users.

A "joint" task is similar to a single-operator task except that it is performed by a number of operators working together. Once again, all operators specified must be available before the task can be started.

An "either" task is performed by one of a specified set of operators. The first operator available to perform the task will perform it. If two or more operators become available at the same time, the operator under least amount of time stress will perform the task. If an operator arrives at an either

task while another operator is performing it, he will skip the task and branch from the node in a normal fashion. If branching is probabilistic, the branch indicating successful completion of the task will be taken by the operator who is not performing the task. The operator who is performing the task will branch according to the outcome of the task.

An "equipment" task has no operators associated with it. Consequently, the only starting condition for an equipment task is that the required number of predecessor completions and the nature of these completions be satisfied. Equipment tasks are unaffected by time stress or other operator characteristics but do affect how much time has been used out of the total amount allowed.

A "cyclic" task is used to provide a delay time until a following task can start. It may or may not have operators associated with it. However, the time to perform a cyclic task is not operator dependent. A cyclic task is designed to end on a specific cycle. Once the precedence requirements for starting the task have been met, the time until the next cycle is determined. The time to perform the task becomes the time remaining to the next cycle. A cyclic task is used in conjunction with a successor task that is required to start on a prescribed cycle.

If an operator is available to perform a task where not all of the predecessor requirements have as yet been met, he must wait for the remainder of the requirements to be satisfied. When this happens, the operator is considered for performance of a gap filler task. A gap filler task is some activity which ought to be performed periodically, time permitting, and is typically executed as a fill-in task during periods where other tasks cannot be executed. Some examples are making instrument checks when the aircraft is on autopilot, taking a coffee break, chatting with the copilot, etc.

There are two conditions that must be met before an operator begins performance of a gap filler task. First, there must be an eligible gap filler task with the specification that this operator may perform it. Second, if other operators are required by the task for which the operator under consideration is waiting, their status must be determined. If any one of the other operators required are presently performing a gap filler task, no gap filler task will be started at this time. However, if this condition is not met, and there is an eligible gap filler task for the operator to perform, he will perform that filler task.

A gap filler task is considered eligible to be performed when its starting conditions are met. There are two conditions upon which eligibility is decided. First, an early start time

is specified for the task. This indicates the earliest possible time the task can be performed for the first time. The second condition involves the time of last performance of the task and the minimum time specified between performances of a gap filler task. Further, once eligibility has been achieved, the task is ranked with all other eligible gap filler tasks based on the task essentiality specified. Gap filler tasks with high values of essentiality will have a higher probability of being performed.

Task Class

A task may be assigned to a particular class, e.g., all switch setting tasks or all visual tasks, etc. This feature is designed to allow counting how often these specific classes of tasks are performed. Any number of tasks may be specified as being in the same task class. In this manner, all tasks that require a special skill or are associated with a particular control or display can be grouped for statistical analysis purposes. These classes have no defined meaning and may be arbitrarily defined for whatever purpose one has for accumulating such statistics.

Task Output

The output side of a SAINT task represents a branching or decision operation. Following completion of a task, a selection is made as to which branches emanating from the task should be activated. The branching type dictates the method by which this selection is made. The five types of branching operations included in SAINT are:

1. Deterministic
2. Probabilistic
3. Conditional, "take first"
4. Conditional, "take all"
5. Modified probabilistic

When a deterministic branching operation is specified, all branches emanating from the task are activated. Thus, the number of requirements for all successor tasks are reduced by one. Essentially, each branch has a probability of one of being selected.

For probabilistic branching, each branch emanating from the task has an associated probability of being selected. Only one of the branches is selected, based on a random number drawn from a uniform zero-one distribution. The sum of the probabilities associated with the branches emanating from a task with a probabilistic output must be 1.0.

For a conditional branching, "take first" operation, each branch is specified with a condition, and the branches are ordered. Each condition is tested in the prescribed order, and the first branch whose condition is satisfied is selected. Conditions may be based on task completions and/or time. The four possible conditions are:

1. specified task completed
2. time less than or equal to specified time
3. specified task not completed
4. time greater than specified time

A branch specified with condition 1 is chosen only if the task specified has been completed prior to the branching operation. For condition 2, the branch is selected if the time into the simulation at the time of branching is less than or equal to the specified time. For condition 3, the branch will be selected if the specified task has not been completed prior to the branching operation. For condition 4, the branch will be taken if the time into the mission is greater than the specified time. The last branch emanating from the task need not have a condition. This branch will be chosen only if none of the other branches have been selected.

A conditional, "take all" branching operation is similar to the conditional, "take first" branching operation. Any of the four conditions may be specified for a branch. In this case the condition on each branch emanating from the task is evaluated and for every condition that is satisfied, the corresponding branch is taken. Once again, the last branch need not have a condition. When this is the case, it is selected only if none of the other conditions on the branches have been satisfied.

Each branch emanating from a task with modified probabilistic branching has associated with it a probability and a probability change. Branching occurs probabilistically but with the probabilities increased or decreased by the prescribed change multiplied by the number of previous completions of the task from which branching is being performed. The initial probabilities must sum to one, while the probability changes must sum to zero. When one of the branches emanating from the task has been decreased in such a manner that its probability of being chosen is zero, branching probabilities are no longer modified.

The branches associated with both probabilistic and modified probabilistic branching must be ordered in such a way that the first branch represents the successful completion of the task. When a task is skipped and branching is probabilistic, only the successful branch will be taken.

Changes to the branching probabilities may also be made due to time stress, operator accuracy, a goal gradient, and

radiation exposure. The effects of time stress, accuracy, and goal gradient on branching probabilities will be discussed in the next section of this report. A discussion of the effect of radiation exposure on the branching probabilities has been published in detail elsewhere (3) and will not be presented in this report.

Additional Task-oriented Concepts

At the time of completion of a task, changes to the network structure and operator assignments can be made. The possible changes are network modification, parameter modification, task clearing, and operator clearing.

Network modification involves the substitution of the characteristics and output side of one task for another task. For example, the completion of task 7 could cause task 10 to replace task 5 in the network. Thus, when task 5 is released, task 10 would be the task that could be started. Branching would then occur from task 10 when it is completed. If task 5 was in progress when the modification took place, branching would occur from task 10 after task 5 is completed. If task 7 was not completed before task 5 was, the outcome of task 5 would dictate the branch(es) taken.

Parameter modification is similar but involves the substitution of one parameter set for another based on some task completion. Once this modification has been performed, the task (or tasks) which previously used the original parameter set to generate performance times will now use the substitute set.

If it is desired to halt a task in progress based on the completion of another task, then a task clearing operation is performed. A task clearing operation halts the ongoing specified task and assumes that the branches emanating from the cleared task are not taken. Further, all operators that were working on the cleared task are set idle.

Another feature of task clearing is the reduction of requirements for release at some other task by what is called a signal operation. This signal operation acts as if a branch has been taken to the signaled task where no branch actually appears in the network.

Operator clearing involves halting the task in progress that is being performed by the specified operator. If the specified operator is cleared from a task, the task itself is cleared, and all operators working on the task are set idle. The signal feature that is part of task clearing may also be specified in the operator clearing operation.

If the operator to be cleared is idle at the time the clearing operation is to be performed, no clearing occurs. However, if the clear and signal feature is specified, the signaling operation will take place. If the operator is waiting for a task to begin, the clearing operation sets him to idle status. Any other operators waiting for the same task are also set idle. The signaling operation, if specified, is then performed.

Statistical Information Collection

SAINT maintains statistics on all tasks and all operators automatically. SAINT also permits the user to specify detailed information for sets of mission iterations. These are described in detail in the SAINT User's Manual (15).

In SAINT there can be multiple source and multiple sink tasks. A source task is a starting point in the network and is the first task to be performed by an operator. A sink task is one which specifies that its completion can cause the completion (realization) of the mission as represented by the network, i.e., one of the sink tasks must be the last task performed in a mission. Since probabilistic branching is part of SAINT, all sink tasks need not be completed in order to complete the mission. Thus, the number of sink tasks (one or more) to complete the mission must be defined. Statistics are collected for each sink task which represent mission performance. In addition, statistics are collected on tasks prescribed by the user. These tasks are called statistics tasks. For all tasks on which statistics are collected, SAINT obtains estimates of the mean, standard deviation, minimum, maximum and a histogram associated with the time of completion of the task. Five types of time statistics are possible.

- F. The time of first completion of the task;
- A. The time of all completions of the task;
- B. The time between completions of the task;
- I. The time interval required to go from the start of a mark task to the completion of the task for which "I" statistics are desired;
- D. The time delay from the first predecessor completion on the task until the task is started, i.e., idle or wait time.

At the completion of a statistics task, task class statistics are collected. For each task class, SAINT obtains estimates of the number of times that tasks of the particular class have been performed prior to the completion of the statistics task. These estimates take the form of the mean, standard deviation, minimum and maximum number of times that tasks of a particular class have been performed.

This completes the discussion of tasks, their characteristics and the method for integrating tasks into a network. In the next section, operator-oriented concepts and their effect on task performance are presented.

SECTION III

OPERATOR-ORIENTED CONCEPTS

Operator characteristics are used in conjunction with the task-oriented concepts to make a mission operator-specific. These characteristics serve to modify the time to perform a task and the probability of successfully completing a task. The characteristics affecting operator performance currently include: speed, accuracy, individual and group stress, and goal proximity. The original development of these characteristics was made by Siegel et. al. (11,12) and the material presented herein is based on their efforts. Extensions have been made to consider cases where more than two operators might be simulated.

The original performance model developed by Siegel and Wolf was dominantly oriented toward the dynamics of workload stress. Initial work on deriving an empirically based relationship between an operationally defined index of time stress and the task parameters (time and success probability) is described elsewhere (13) as are the attempts to compare model results to field observations (12). No attempt is made here to critique either of these efforts nor to modify any of the proposed constructs or relationships. The basic concepts and techniques used by Siegel and Wolf in implementing accuracy, group stress, and goal proximity dynamics have been incorporated into the SAINT computer program. However, it should be mentioned that no effort has been expended as yet, by Siegel or others, to revalidate these proposed refinements to performance models. While such work is obviously necessary, it was beyond the scope of the present effort, which was intentionally restricted to developing general programs for analyzing network models of operator and task dynamics.

Speed

Speed is intended to reflect one aspect of skill proficiency and is an operator attribute that directly affects task performance time or task duration. The inclusion of a speed factor in SAINT allows for the simulation of operators who might be faster or slower than an average operator.

The average operator is assigned a speed factor of unity. Faster or more proficient operators are assigned a speed factor that is less than unity. Their task times are multiplied by this factor and thereby reduced. A speed factor that is greater than unity indicates a less proficient operator that is slower than average. For example, if a task took 3 seconds on the average but the operator was presumed to be 10 percent less proficient than average, he would be assigned a speed factor of

1.1 and for him, the average task time would be 3.3 seconds (3.0 x 1.1).

For a single operator task, an either task or a gap filler task, the speed factor for the task, f_t , is the speed factor for the operator performing the task.

If there is more than one operator performing a task (i.e., a joint task), a single speed factor is computed in order to modify the task performance time. It is assumed that a task can only be performed as quickly as the slowest operator engaged in performing it. Thus, the speed factor for a task involving multiple operators corresponds to the slowest operator working on the task, and is calculated as

$$f_t = \max_{j \in J(t)} (f_j)$$

where

f_j is the speed factor for operator j ,

$J(t)$ is the set of operators performing task t

and f_t is the speed factor for task t .

A time to perform the task is generated using the distribution type and parameter set number associated with the task. If necessary, this task duration is also modified by the adjustment factor and to account for radiation and stress effects. Before these other modifications are made, the speed factor is applied to make the task duration operator specific; that is,

$$T'_t = T_t \cdot f_t$$

where

T_t is the task duration before operator speed is considered for task t ,

f_t is the task speed factor,

and T'_t is the task performance time with the task speed factor applied.

When an equipment task is being performed, no modification for speed is made to task duration as no operators are associated with this task type. Further, no speed adjustment is made to the task duration of a cyclic task, even if operators are associated with it, due to the nature of this task type.

Accuracy

Accuracy is the other aspect of skill proficiency that is explicitly treated and this effect is included in the success probability calculation to allow for operators who might be less accurate or more accurate than an average operator. The average operator has an accuracy value of unity. Those operators who are less accurate than average are assigned an accuracy factor in the range of 1.0 to 1.2. The more accurate operator has an accuracy factor in the range 0.8 to 1.0.

Upon input, the operator accuracy value is transformed so that the average operator has an accuracy factor of zero. The less accurate operator is assigned an accuracy less than zero, while the more accurate operator has an accuracy greater than zero. This transformation is performed by the operation:

$$a_j' = 5(1 - a_j)$$

where

a_j is the assigned accuracy factor for operator j

$$(0.8 \leq a_j \leq 1.2),$$

and a_j' is the transformed accuracy factor for operator j

$$(-1.0 \leq a_j' \leq 1.0).$$

The transformed accuracy becomes the accuracy factor associated with the operator and is used in all calculations involving operator accuracy.

If an operator is performing a task alone (i.e., a single operator task or an either task), the accuracy value of that operator is used to modify the probability of successfully performing the task. If there is more than one operator working on the task (i.e., a joint task), the accuracy used to determine the probability of success is the value associated with the least accurate operator. Thus, the calculation of the task accuracy is

$$a_t = \min_{j \in J(t)} (a_j')$$

where

$J(t)$ is the set of all operators performing task t ,

and a_t is the task accuracy factor.

A probability of success is associated only with those tasks which branch using either the probabilistic or modified probabilistic method. For these tasks, the inputted probability of success goes through a series of potential modifications.

The order in which the probability of successfully completing a task is modified is as follows: modified probabilistic effects; accuracy factor; goal gradient effects; stress effects and radiation effects. The equations for modifying the probability of success due to operator accuracy are:

$$p'_t = \begin{cases} p_t(a_t + 1) & ; a_t \leq 0 \\ p_t + (1-p_t)a_t & ; a_t > 0 \end{cases}$$

where

p_t is the probability of success for task t before accuracy of the operator is considered,

and p'_t is the modified probability of success for task t .

The probabilities of taking branches other than the successful branch are transformed in proportion to the change in the success probability. Figure 1 illustrates the probability of success modification for different task accuracy factors.

On equipment tasks and cyclic tasks, no operator accuracy is considered due to the nature of these task types. On gap filler tasks, no branching operation is allowed. Thus, no accuracy calculation is made.

Stress

The workload stress on an operator is viewed as a time pressure imposed on him by a discrepancy between the amount of work to be done and the time remaining for doing it. The effects of this workload stress are reflected in the operator's task duration and task success. The manner in which the stress is calculated and the manner in which it affects performance will now be described.

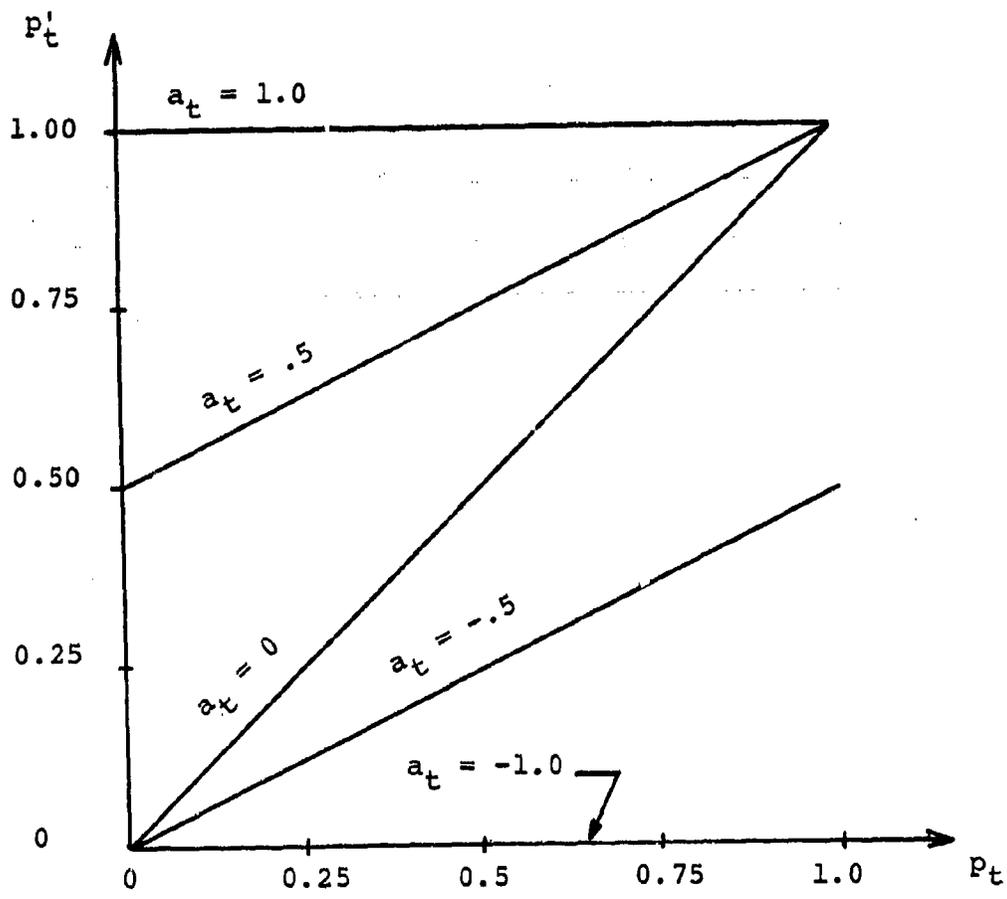


Figure 1. Probability of Success Modification for Different Task Accuracy Factors

Prior to the start of a task in which operator characteristics are to be considered (i.e., single operator tasks, joint tasks, and either tasks), the status of each operator involved in performing the task is assessed. The status is based on the amount of time the operator has available to complete his assigned series of tasks over the remaining portion of the mission, the average amount of essential time and non-essential time required to finish all remaining tasks in the mission, and the operator's speed factor. There are thus three time values to be considered: 1) time used, 2) time available, and 3) time remaining. Time remaining is derived by calculating the difference between time available and time used. Status, defined later, will be based on time remaining. The SAINT user assigns each operator an allotment of time for performing his portion of the mission. Given this value for time available, the remaining time available is calculated by considering the time requirements for all essential and non-essential tasks from the one now being considered to the task that ends the mission.

The essential time and non-essential time remaining for each operator from the start of each task to mission completion is either prescribed by the user or calculated through a series of benchmark iterations. These benchmark values are derived by simulating the task network without any practical limit on the allotted time available for completing these tasks. In such benchmark iterations, the operators are not constrained, which then permits an estimate of the amount of time necessary for them to finish their remaining tasks irrespective of workload conditions or other environmental factors which dynamically modify performance. The benchmark values provide a static baseline of idealized performance times for task completion under benign conditions. As the program stands now, all performance dynamics captured by the Siegel-Wolf model are oriented toward modifying this idealized representation to reflect the realities of an operator's response to contingency events, workload constraints and environmental variables which may degrade performance.

The status of the operator is now defined to be in one of three states which are determined by the following conditions:

$$\text{State 1: } f_j (E_{tj} + N_{tj}) \leq TM_j - U_t$$

$$\text{State 2: } f_j E_{tj} \leq TM_j - U_t < f_j (E_{tj} + N_{tj})$$

$$\text{State 3: } f_j E_{tj} > TM_j - U_t$$

where

E_{tj} is the essential time remaining for operator j at task t ,

N_{tj} is the non-essential time remaining for operator j at task t ,

U_t is the time at which task t is to be started, i.e., the accumulated time used on preceding tasks

TM_j is the assigned total time available for operator j to complete his portion of the mission,

and f_j is the speed factor for operator j

Thus $TM_j - U_t$ is the calculated value for time remaining.

If E_{tj} and N_{tj} are not provided as input data by the SAINT user, they are calculated from the benchmark iterations. If the operator is in State 1, he has "sufficient" time to perform task t . If the operator is in State 3, he does not have "sufficient" time to perform task t . If he is in State 2, he has "sufficient" time to perform all essential tasks remaining but not all non-essential tasks remaining. Note that the criterion of sufficiency is assumed to be a function of the average of the performance times.

The work load stress for each operator (referenced to his completing the mission) is calculated based on the above states. If an operator is in state 1 or 2, his stress is equal to unity (no stress). If the operator is in state 3, his stress (based on time to mission completion) is

$$s_{tj}(z) = \max [1; \min [\frac{E_{tj}}{TM_j - U_t}; 5.0]]$$

where

$s_{tj}(z)$ is the stress on operator j at the start of task t based on mission completion at task z .

Many missions consist of a series of segments or phases. Often the operations in the segment are dominantly geared to achieving the goal of that phase, relatively independent of other subsequent phases yet to be completed. For example, if

an aircraft has several weapons, the release of each may be important by itself, and it seems reasonable to postulate that an operator's time stress in a weapon delivery is governed by the time available for that mission phase rather than the time available for the total mission. The end points for such mission phases may then be taken as intermediate points where time stress applies, and the dynamics of the operator's response to workload need to reflect such considerations.

The time available for completing each phase is assigned by the SAINT user. These allotments may be different for each operator and are specific to each phase. It should be pointed out that these time allotments are meant to reflect externally imposed constraints on mission duration and are not directly related to the task times. This allotment remains fixed. The accumulated time used depends on the variability in the operator's performance, how much time was lost on failed tasks, and other factors. Time remaining thus shrinks as a consequence of the operator's response to conditions encountered.

Given the assigned time available to the operator for reaching these intermediate points, the state of each operator (relating to the next intermediate stress point in his task sequence) is calculated as:

$$\text{State 1: } f_j((E_{tj} - E_{nj}) + (N_{tj} - N_{nj})) \leq TI_{nj} - U_t$$

$$\text{State 2: } f_j(E_{tj} - E_{nj}) \leq TI_{nj} - U_t < f_j((E_{tj} - E_{nj}) + (N_{tj} - N_{nj}))$$

$$\text{State 3: } f_j(E_{tj} - E_{nj}) > TI_{nj} - U_t$$

where

n is the next intermediate stress task for operator j,

and TI_{nj} is the time available for operator j to reach task n.

The intermediate stress on each operator is calculated based on the above states. If an operator is in state 1 or 2, his intermediate stress is equal to unity. If an operator is in state 3, his intermediate stress is calculated as:

$$s_{tj}(n) = \max [1; \min [\frac{E_{tj} - E_{nj}}{TI_{nj} - U_{tj}} ; 5.0]]$$

where

$s_{tj}(n)$ is the stress on operator j at the start of task t based on his next intermediate stress task.

For each operator associated with task t , an overall operator stress s_{tj} is determined as:

$$s_{tj} = \max (s_{tj}(z), s_{tj}(n)).$$

Thus if mission stress is larger than segment (intermediate) stress, intermediate stress is not used, but if the intermediate stress is the larger of the two, it is used instead of mission stress.

For the task under consideration, a task stress is determined as the maximum of the overall stress values of the operators performing the task:

$$s_t = \max_{j \in J(t)} (s_{tj})$$

where

s_t is the stress for task t ,

and $J(t)$ is the set of operators performing task t .

The empirical literature on stress typically portrays its impact as being an organizing influence on behavior for low values of stress and a disorganizing influence for high values. Siegel and Wolf portrayed the organizing influence of workload stress as an improvement in task performance exhibited by a decrease in the mean and standard deviation of task times and an increase in the probability of task success. Correspondingly, the disorganizing influence of stress was implemented as a decrease in the probability of success and an increase in the mean and standard deviation of task times. Assuming the functional relationships between stress and these task attributes is given, the question becomes one of deciding which one to use; the organizing or disorganizing representation. To distinguish between these, Siegel and Wolf utilize an operator stress threshold, M_j . If stress assumes a value equal to or less than M_j , stress is an organizing influence, but if stress exceeds M_j , the influence is disorganizing.

Once the task stress has been determined, an additive stress factor is calculated. This additive stress value accounts for stress on other operators who might affect the performance of those operators working on task t. Consequently, for each operator not involved in the performance of task t, an additive stress factor is calculated. If an operator not performing task t has a stress of one (the minimum allowed), his additive stress value is zero. If his stress is between one and his threshold (stress improves task performance), the additive stress value is determined as

$$A_t = \max_{j \notin J(t)} \left[\frac{s_{.j} - 1}{M_j - 1} \right] ; 0 \leq A_t \leq 1$$

where

A_t is the additive stress for task t,

and $s_{.j}$ is the stress on operator j at his present mission location.

Note that the ratio indicated in the above equation is greater than one anytime stress ($s_{.j}$) is above threshold (M_j). This implies that once any operator not involved in the task exceeds his stress threshold, the stress added to the group doing the task is at a maximum value (1.0). Depending on the current value of stress for various members of the group doing the work and their corresponding stress thresholds, it is possible that this augmented stress could also induce their exceeding threshold. Consequently, once a part of the crews' behavior reaches the point of disorganization, disorganized behavior may be induced in others. While this appears to capture the nature of group dynamics under stress, continued use of the model and empirical tests will illuminate the validity of this formulation and hopefully suggest other approaches if this one proves inappropriate.

The total task stress S_t , is the sum of the task stress s_t , and the additive stress, A_t , i.e., $S_t = s_t + A_t$. Then S_t is used to modify the task duration and the success probability for the task.

To determine the effect of stress on task performance, a task threshold level, M_t , is calculated as the threshold level of the operator under the greatest stress, i.e.,

$$M_t = M_k$$

where

$$k \text{ is the member of } J(t) \text{ such that } s_{tk} = \max_{j \in J(t)} (s_{tj})$$

Joint tasks are those whose completion time is governed by the group doing the work, and therefore the task ends whenever the slowest member's performance contribution to that task is completed. While all the foregoing discussion has centered on the group dynamics of stress, it is also necessary to consider what threshold will be used to determine whether the group stress is an organizing or disorganizing influence. The hypothesis postulated here is that performance of the group will be determined by the member who is most stressed. If his stress threshold is high, he may prevent another member's lower threshold from disrupting the joint task performance; but if his threshold is low and his stress high, then his disorganized behavior will establish the group's activity. Again, this representation appears reasonable and consistent with the nature of a joint task, but further empirical test and evaluation is indicated. The effect of stress on task performance is a function of the total task stress, S_t , and the task threshold level, M_t . These effects will now be discussed.

Skipping of tasks due to stress. When $S_t > 1$, task t may be skipped if its essentiality, e_t , is less than 1.0. It is presumed that the probability of skipping a task is related to e_t , s_t and M_t . Let y_t be a deviate drawn from a normal distribution with mean μ , and standard deviation, σ .

$$\text{where } \mu = \left[\frac{S_t - 1}{M_t - 1} \right]$$

$$\text{and } \sigma = 0.15.$$

The rule for skipping task t is:

if $y_t \leq e_t$, task t will be performed;

and if $y_t > e_t$, task t will be skipped.

The probability that task t is skipped is given by

$$P[\text{task } t \text{ is skipped}] = p[y_t > e_t] = \frac{1}{\sqrt{2\pi}} \int_z^\infty e^{-w^2/2} dw$$

where $z = (e_t - \mu)/\sigma$.

It is seen that smaller values of $(e_t - \mu)$ yield higher probabilities of skipping a task for a given value of σ .

This particular formulation has no empirical basis, and it may be of interest, therefore, to examine more carefully the manner in which the operation is performed. First, the value for the mean (μ) will vary not only as stress (S_t) varies but as the stress threshold (M_t) varies. Second, this stress threshold depends on the members of the group performing the task, if it is a joint task. Consequently, to make the demonstration simple, it will be assumed that the task involves a single operator. Further, since a stress threshold value of 2.3 has most often been used in previous simulations, that value will be used in calculations here.

Table I shows the z values corresponding to different stress levels for each level of essentiality. Figure 2 illustrates how the probability of skipping a task increases with stress for each of 10 non-zero essentialities. An increase in the value of M_j (or correspondingly, in M_t) will shift these curves to the right and a decrease in M_t will shift them to the left, such that for a lower stress threshold increasing stress will more often lead to an operator's skipping non-essential tasks.

Effect of Stress on Time to Perform Task t. Stress affects the task duration depending on the total task stress value. Let T'_t be the task duration after stress is considered and T_t the task duration prior to stress consideration. If $S_t < M_t$, then:

$$T'_t = T_t (-1.829 x^3 + 3.7422 x^2 - 2.35075 x + 1)$$

where

$$x = \frac{S_t - 1}{M_t - 1}$$

If $M_t \leq S_t \leq M_t + 1$, then:

$$T'_t = (2S_t - 2M_t + 1) T_t - (S_t - M_t) m_t$$

where

m_t is the mean of the task performance time distribution.

If $S_t > M_t + 1$, then:

$$T'_t = 3T_t - m_t .$$

The above equation forms are equivalent to those used in the Siegel-Wolf model (8), the first being empirically derived (a least squares regression analysis of a laboratory study of time stress) whereas the others are conceptual in nature.

Effect of Stress on Probability of Success. Stress also affects the probability of successfully completing a task, p_t , depending on the total task stress value. If $S_t \leq M_t$, then

$$p'_t = \frac{p_t (M_t - S_t) + S_t - 1}{M_t - 1} .$$

If $M_t < S_t \leq M_t + 1$, then

$$p'_t = (p_t - 1) (S_t - M_t) + p_t$$

If $S_t > M_t + 1$, then

$$p'_t = (.5)p_t$$

If the modified success probability, p'_t , becomes greater than 1, it is set to 1. If it drops below 0, it is set to zero. The probabilities of taking branches other than the successful branch are modified in proportion to the success probability modification. Once again, the equation forms used in SAINT for the success probability modification are equivalent to those used in the Siegel-Wolf model (12).

TABLE I
 z VALUES AS A FUNCTION OF STRESS AND ESSENTIALITY
 Essentiality, e_t

STRESS S_t	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
1.00	0.67	1.33	2.00	2.67	3.33	4.00	4.67	5.33	6.00	6.67
1.10	0.15	0.82	1.49	2.15	2.82	3.49	4.15	4.82	5.49	6.15
1.20	-0.36	0.31	0.97	1.64	2.31	2.97	3.64	4.31	4.97	5.64
1.30	-0.87	-0.21	0.46	1.13	1.79	2.46	3.13	3.79	4.46	5.13
1.40	-1.38	-0.72	-0.05	0.62	1.28	1.95	2.62	3.28	3.95	4.62
1.50	-1.90	-1.23	-0.56	0.10	0.77	1.44	2.10	2.77	3.44	4.10
1.60	-2.41	-1.74	-1.08	-0.41	0.26	0.92	1.59	2.26	2.92	3.59
1.70	-2.92	-2.26	-1.59	-0.92	-0.26	0.41	1.08	1.74	2.41	3.08
1.80	-3.44	-2.77	-2.10	-1.44	-0.77	-0.10	0.56	1.23	1.90	2.56
1.90	-3.95	-3.28	-2.62	-1.95	-1.28	-0.62	0.05	0.72	1.38	2.05
2.00	-4.46	-3.79	-3.13	-2.46	-1.79	-1.13	-0.46	0.21	0.87	1.54
2.10	-4.97	-4.31	-3.64	-2.97	-2.31	-1.64	-0.97	-0.31	0.36	1.03
2.20	-5.49	-4.82	-4.15	-3.49	-2.82	-2.15	-1.49	-0.82	-0.15	0.51
2.30	-6.00	-5.33	-4.67	-4.00	-3.33	-2.67	-2.00	-1.33	-0.67	0.00
2.40	-6.51	-5.85	-5.18	-4.51	-3.85	-3.18	-2.51	-1.85	-1.18	-0.51
2.50	-7.03	-6.36	-5.69	-5.03	-4.36	-3.69	-3.03	-2.36	-1.69	-1.03
2.60	-7.54	-6.87	-6.21	-5.54	-4.87	-4.21	-3.54	-2.87	-2.21	-1.54
2.70	-8.05	-7.38	-6.72	-6.05	-5.38	-4.72	-4.05	-3.38	-2.72	-2.05
2.80	-8.56	-7.90	-7.23	-6.56	-5.90	-5.23	-4.56	-3.90	-3.23	-2.56
2.90	-9.08	-8.41	-7.74	-7.08	-6.41	-5.74	-5.08	-4.41	-3.74	-3.08

TABLE 1 Cont'd

STRESS S _t	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
3.00	-9.59	-8.92	-8.26	-7.59	-6.92	-6.26	-5.59	-4.92	-4.26	-3.59
3.10	-10.10	-9.44	-8.77	-8.10	-7.44	-6.77	-6.10	-5.44	-4.77	-4.10
3.20	-10.62	-9.95	-9.28	-8.62	-7.95	-7.28	-6.62	-5.95	-5.28	-4.62
3.30	-11.13	-10.46	-9.79	-9.13	-8.46	-7.79	-7.13	-6.46	-5.79	-5.13
3.40	-11.64	-10.97	-10.31	-9.64	-8.97	-8.31	-7.64	-6.97	-6.31	-5.64
3.50	-12.15	-11.49	-10.82	-10.15	-9.49	-8.82	-8.15	-7.49	-6.82	-6.15
3.60	-12.67	-12.00	-11.33	-10.67	-10.00	-9.33	-8.67	-8.00	-7.33	-6.67
3.70	-13.18	-12.51	-11.85	-11.18	-10.51	-9.85	-9.18	-8.51	-7.85	-7.18
3.80	-13.69	-13.03	-12.36	-11.69	-11.03	-10.36	-9.69	-9.03	-8.38	-7.69
3.90	-14.21	-13.54	-12.87	-12.21	-11.54	-10.87	-10.21	-9.54	-8.87	-8.21
4.00	-14.72	-14.05	-13.38	-12.72	-12.05	-11.38	-10.72	-10.05	-9.38	-8.72
4.10	-15.23	-14.56	-13.90	-13.23	-12.56	-11.90	-11.23	-10.56	-9.90	-9.23
4.20	-15.74	-15.08	-14.41	-13.74	-13.08	-12.41	-11.74	-11.08	-10.41	-9.74
4.30	-16.26	-15.59	-14.92	-14.26	-13.59	-12.92	-12.26	-11.59	-10.92	-10.26
4.40	-16.77	-16.10	-15.44	-14.77	-14.10	-13.44	-12.77	-12.10	-11.44	-10.77
4.50	-17.28	-16.62	-15.95	-15.28	-14.62	-13.95	-13.28	-12.62	-11.95	-11.28
4.60	-17.79	-17.13	-16.46	-15.79	-15.13	-14.46	-13.79	-13.13	-12.46	-11.79
4.70	-18.31	-17.64	-16.97	-16.31	-15.64	-14.97	-14.31	-13.64	-12.97	-12.31
4.80	-18.82	-18.15	-17.49	-16.82	-16.15	-15.49	-14.82	-14.15	-13.49	-12.82
4.90	-19.33	-18.67	-18.00	-17.33	-16.67	-16.00	-15.33	-14.67	-14.00	-13.33
5.00	-19.85	-19.18	-18.51	-17.85	-17.18	-16.51	-15.85	-15.18	-14.51	-13.85

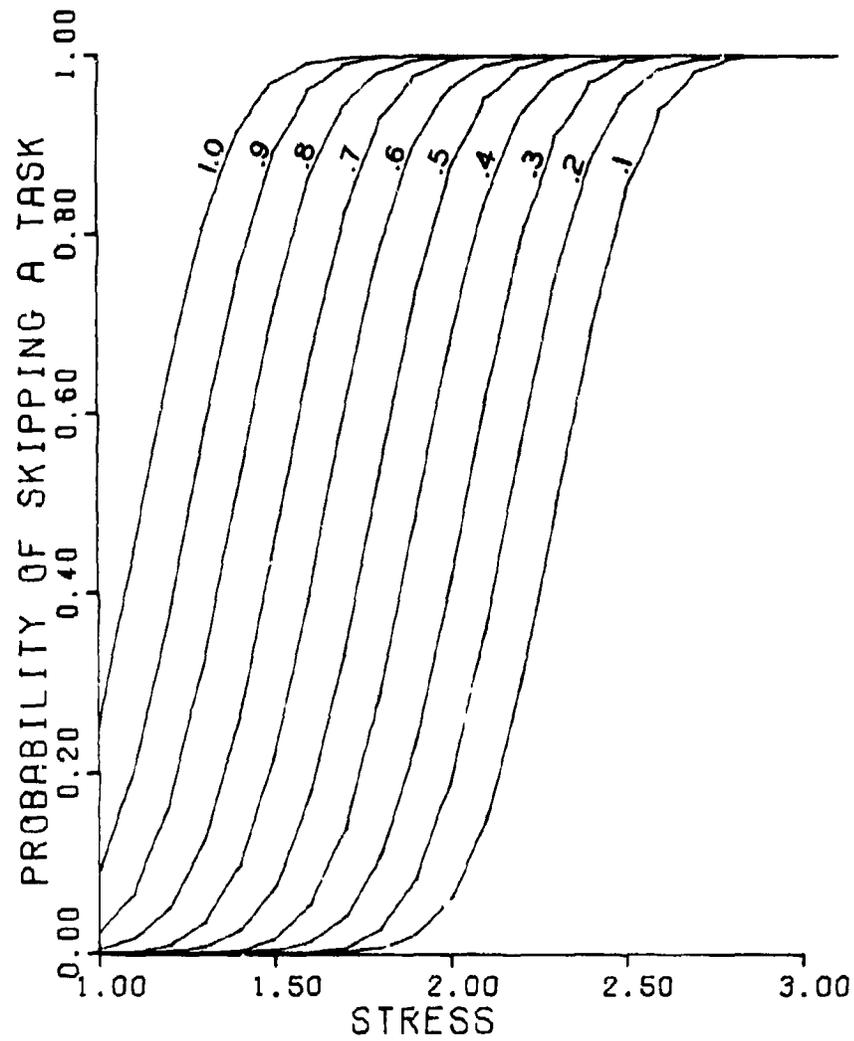


Figure 2. Probability of Skipping a Task as a Function of Stress and Essentiality

Goal Gradients

The goal gradient effect or so-called "end spurt" is a concept which represents an increase in performance caused by the proximity to a goal or end point. In the Siegel-Wolf model as in SAINT, this is modeled through an increase in the probability of success for a specified percentage of the later tasks in a mission segment that have a stress equal to unity (no stress). The goal gradient value prescribed for each operator indicates the percentage of mission completion required before the goal gradient may take effect. This percentage may be no less than 75 per cent.

The task goal gradient value is set to the highest value associated with any operator performing the task. The goal gradient effect is shown in tabular form in Table II. The goal gradient augmentation, g_t , for the task is added to the success probability to arrive at a new probability of success as:

$$p'_t = p_t + g_t .$$

If the SAINT user believes this construct does not apply to the situation being represented, a goal gradient value of zero may be used, indicating this additive factor is to be considered only when the mission is completed, i.e., effectively not considered at all. Further, the value obtained for p'_t is not allowed to exceed 1.0, so if p_t is close enough to 1.0 the impact of g_t may be less than indicated by the tabled values, but in that case virtually perfect performance is assured, subject to other manipulations occurring subsequent to considering the goal gradient.

Table II
Goal Gradient Additive Factor

<u>Per Cent Mission Completed</u>	<u>Success Probability Augmentation g_t</u>
75.0	0
77.5	0.004
80.0	0.008
82.5	0.012
85.0	0.017
87.5	0.022
90.0	0.028
92.5	0.036
95.0	0.045
96.0	0.051
97.0	0.059
98.0	0.069
99.0	0.082
100.0	0.100

Cohesiveness

Cohesiveness is a stress-oriented statistic collected on each task for which task stress calculations are made. The index of cohesiveness is an attempt to measure the joint stress condition of the entire mission crew. It is only an indicator and is not used to modify mission performance.

For each operator not involved in the performance of task t , an index of cohesiveness is calculated as follows:

$$c_j = \frac{S_t s_{.j}^{-1}}{M_t M_j^{-1}} \quad \text{for } j \notin J(t).$$

The index of cohesiveness for the task is then assigned the largest index of any operator not associated with the task, i.e.,

$$C_t = \max_{j \notin J(t)} (c_j)$$

As can be seen, if the task stress and the stress on the selected external operator are both 1 (no stress condition), the value of the index is 0. When both stresses are equal to their corresponding thresholds, the index value is 1. Thus, the lower the index value, the greater the team harmony, or cohesion.

Computational Sequence

Basically, two dependent variables are used to define task performance: task duration and probability of successfully completing the task. Throughout this section, the independent variables were presented. For each relation, the modifications involved in the dependent variables was specified in terms of a previously computed value for the dependent variable. The order in which these modifications are made, i.e., the computational sequence, is shown below:

Task Duration

1. Speed Factor (as a function of the proficiency of team members).
2. Adjustment Factor (as a function of number of task completions).
3. Workload Stress Considerations (as a function of time remaining versus time required).
4. Radiation Effects (as a function of the duration and level of exposure).

Probability of Success

1. Modified Probabilistic Branching Effects (if applicable to this task).
2. Accuracy Factor (as a function of the proficiency of team members).
3. Goal Gradient (if deemed appropriate and applicable at this stage into the mission).
4. Stress Considerations (as above).
5. Radiation Effects (as above).

This completes the presentation of operator-oriented concepts and how they affect task performance. A modular design has been maintained in SAINT to allow for different operator characteristics and for the adaptation and extension of the concepts. This flexibility should enhance the use of SAINT and allow different users to refine their representation of operator and group dynamics as they accumulate empirical data. In the next section, the graphical concepts that are used to portray a SAINT model are presented.

SECTION IV

SAINT MODELING

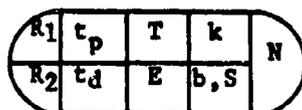
A four-step procedure is recommended when using SAINT to model and analyze a mission composed of tasks and operators. The first step is to use the SAINT concepts and symbolism to develop a network model. The second step is to translate the network model into a data input set for the SAINT program. This is accomplished by using the SAINT input description. The third step is the simulation of the network by the SAINT program. The final step is to analyze the statistical results obtained from the SAINT program. Steps 2, 3, and 4 are discussed in the SAINT User's Manual (15). The discussion in this section will concentrate on modeling and the graphical symbolism that is an integral part of SAINT.

The symbols to model a mission using SAINT are illustrated in Figure 3. The requirements for starting a task are shown on the input side of a node. The number of requirements for starting a task the first time is on the top (R_1) and the requirements for starting a task after the first time are shown on the bottom (R_2). The description of the task is graphically portrayed by a rectangle which is subdivided into six squares. In each square, a value or symbol is placed to indicate the type of task, the features which determine the time required to perform the task, and the statistical information to be collected at the task. The output side of a node contains the task number which is the node label. The shape of the output side of the node indicates the type of branching to occur from the node. Branching from a task is the means by which potential successor tasks are identified.

The number of requirements for starting a task is specified on the input side of the node. Shading on the input side indicates that each requirement must be from a different predecessor task, for both first and subsequent task release.

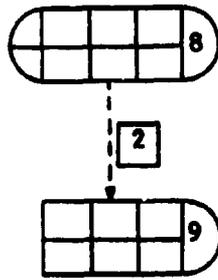


Network modification involves the substitution of the task description and output side of one task for those of another task.



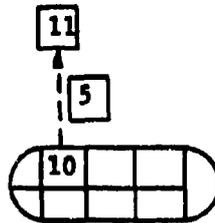
- N task number
- R_1 number of releases required for first release of task N
- R_2 number of releases required for subsequent release of task N
- t time descriptors:
- t_p parameter set number
 - t_d distribution type
- T task type:
- S single operator task
 - J joint task
 - E either task
 - Q equipment task
 - C cyclic task
 - F gap filler task
- E task essentiality ($0 \leq E \leq 1$)
- k task class
- b task adjustment factor
- S task involved in data collection such as a mark node (S=M) or a statistics node (S = F, A, B, I, or D)

Figure 3. Task Symbol and Terminology



The above illustration indicates that task 9 will be substituted for task 8 upon completion of task 2.

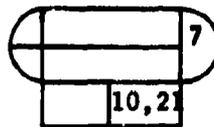
Parameter modification involves the substitution of one parameter set for another based on a task completion.



The above illustration indicates that parameter set 10 is replaced by parameter set 11 upon completion of task 5.

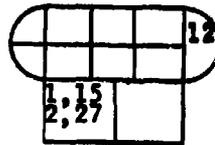
Below the central (task description portion of the task, a rectangle is placed if a clearing operation is to take place upon completion of the task under which it appears.

The right-hand portion of the rectangle is used if other tasks can be interrupted or cleared while in progress upon completion of the task under which it appears. When a task is cleared, SAINT allows the number of requirements for other tasks to be reduced. This allows the starting of other tasks in the network when a task is halted in progress. In addition, the operator (or operators) performing the cleared task is made available and can be reassigned to another task in the same manner as when a task is completed. First consider the clearing of a task.



The task symbol above indicates that task 10 is to be halted in progress when task 7 is completed. In addition, once the clearing of task 10 has taken place, the number of requirements for task 21 is reduced by one.

The left-hand portion of the lower rectangle is used if operators are to be cleared upon task completion. If the operator to be cleared is presently performing a task, then the entire task is cleared. All operators performing the task are made available for reassignment to another task, as in task clearing. If the operator to be cleared is waiting for a task to start, then all operators waiting for the task are made available for reassignment. If the operator to be cleared is not assigned to a task, no clearing need be performed. When an attempt is made to clear an operator, SAINT allows the number of requirements for other tasks to be reduced, as in task clearing. The clearing of operators 1 and 2 is indicated by the following symbol:



This symbol indicates that operator 1 is to be cleared when task 12 is completed, followed by the reduction of the number of requirements for task 15 by one. In addition, operator 2 is to be cleared and the requirements for task 27 are to be reduced by one.

Branch Symbolism

The branches of the network are the lines between the nodes of the network. The parameters of a branch depend on the type of branching being performed. The symbols used for each branching type are shown in Figure 4. The number of parameters and the parameter definitions for each branching type are given below:

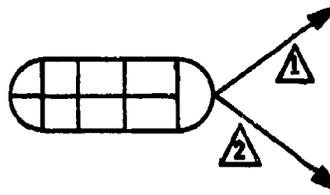
Branching Type	Number of Parameters	Parameter Definition	
		1	2
Deterministic	0	-	-
Probabilistic	1	Probability	-
Conditional, Take First	2	Condition	Value
Conditional, Take All	2	1. specified task completed	1. task number
		2. time less than or equal to specified time	2. specified time
		3. specified task not completed	3. task number
		4. time greater than specified time	4. specified time
Modified Probabilistic	2	Probability	Change in probability

Examples of the parameter specifications are shown in Figure 4. Below the branch, a number in a triangle is used to indicate the operator or operators traversing the branch. In some cases, the operator number will not be indicated as it is clear from the network which operator is traversing the branch.

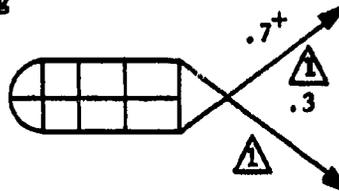
Operator Symbolism

Operators flow through the network and, in so doing, perform tasks. Operator or operators are identified with branches by placing a triangle below the branch with the operator number in the triangle. There is no need or requirement to indicate the operator performing a task as this can be determined from the incoming branches to the task and the task type. SAINT automatically maintains operator status and the tasks that have been released to which the operator can be assigned. Operators are assigned to gap filler tasks on the basis of task essentiality.

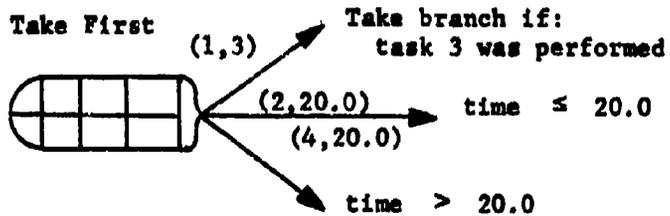
1. Deterministic Branching



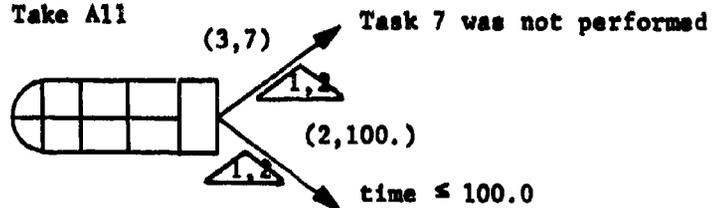
2. Probabilistic Branching



3. Conditional Branching, Take First



4. Conditional Branching, Take All



5. Modified Probabilistic Branching

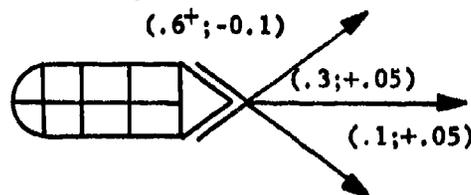
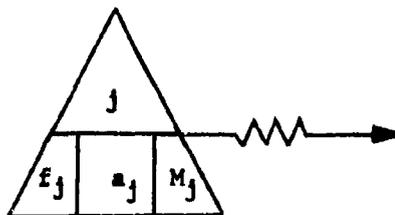


Figure 4. Branching Types, Symbols and Terminology

The characteristics of an operator are illustrated graphically as shown below:



where j is the operator number or label,

f_j is the speed factor for operator j ,

a_j is the accuracy factor for operator j ,

and M_j is the stress threshold for operator j .

The initial task to which the operator is assigned is pointed to by the above symbol. The operator symbol and pointer are not considered to be a node or branch of the network.

Examples

The examples presented herein will be kept small so that the modeling features contained within SAINT can be demonstrated. Throughout the examples, only the pertinent information regarding the feature being demonstrated will be included. No attempt will be made to illustrate all the features available within SAINT. In fact, the discussion will concentrate on the network modeling features since general discussions of the operator and task parameters have been discussed elsewhere.

No attempt is made to rationalize the examples presented by suggesting the kinds of task situations where the representation described would be useful. This is left to the imagination of the user. Often, the best approach is to start with a simple diagram, not using all of the symbols at first but simply laying out the path of normal activity, assuming no contingency events or other complications occur. As this is done, areas believed to require special treatment later can be marked to call attention to the fact that the symbols might need review and modification. One of the more common mistakes made is to try being too detailed in the first attempts to use the symbols. As experience is gained, the usefulness of the symbols and features of SAINT will become more apparent, but this requires discovery through application. Hopefully, the following discussion is adequate to suggest the

general nature of the symbols and ways in which they might be used. The treatment is neither unique to any specific use nor exhaustive of the uses one may have for combining the symbols.

In Figure 5, a series of tasks to be performed by a single operator is shown. The triangle at the beginning of the network indicates that operator 1 is to start at task 2. Operator 1 has a speed factor of 0.9, an accuracy factor of 1.0, and a stress threshold of 2.5. Task 2 is a single operator task and a source task. Source tasks are indicated by having zero requirements for their first release, ($R_1 = 0$). An adjustment factor of 0.8 is associated with task 2. Branching from task 2 is probabilistic with the branch from 2 to 3 being the successful branch. This is indicated with a plus sign attached to the probability of going from node 2 to node 3. The probability that the operator did not successfully accomplish the task is 0.1 and is indicated on the feedback branch around task 2. Note that for subsequent releases of task 2, one requirement is specified. If task 2 is failed, it is performed again, and the time to perform it is modified by the adjustment factor (0.8). This factor would continue to be applied for each subsequent failure and re-execution of task 2. In both branches emanating from task 2, operator 1 is assigned, and a small triangle with a 1 inside of it is placed beneath both branches emanating from task 2. Task 3 is a single operator task similar to task 2. For task 3, branching is modified probabilistic which indicates that there is a changing probability associated with the branching from task 3. In this case the probability of successfully completing the task is .5 which would result in operator 1 being sent to task 4. Each time task 3 is failed and then repeated, this success probability is increased by 0.1. Task 4 represents the last task of the network and statistics describing task 4 would relate to mission or network performance.

The next example involves two operators who are required to complete their mission in 20 hours. The network representing the mission is shown in Figure 6. Operator 1 starts at task 1 and operator 2 at task 2. Following tasks 1 and 2, operators 1 and 2 must work together on task 3. Task 3 is a joint task ($T = J$) and has two requirements before it can be started, one for each operator. Branching from task 3 is conditional, take first. The branch from 3 to 4 is taken if the current time is less than or equal to 20 (condition 2). Both operators 1 and 2 are then transferred to task 4 which represents the mission being completed on time. If the time of completion of task 3 is greater than 20 (condition 4) then branching to task 5 occurs and the mission is not completed on time. In an enlarged treatment of a mission, other activities could then represent what the crew does from this point on: re-execute an attack, go home, etc.

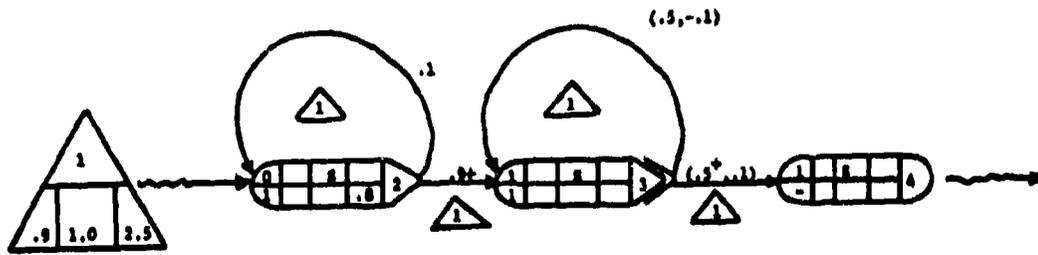


Figure 5. A Series of Tasks by a Single Operator

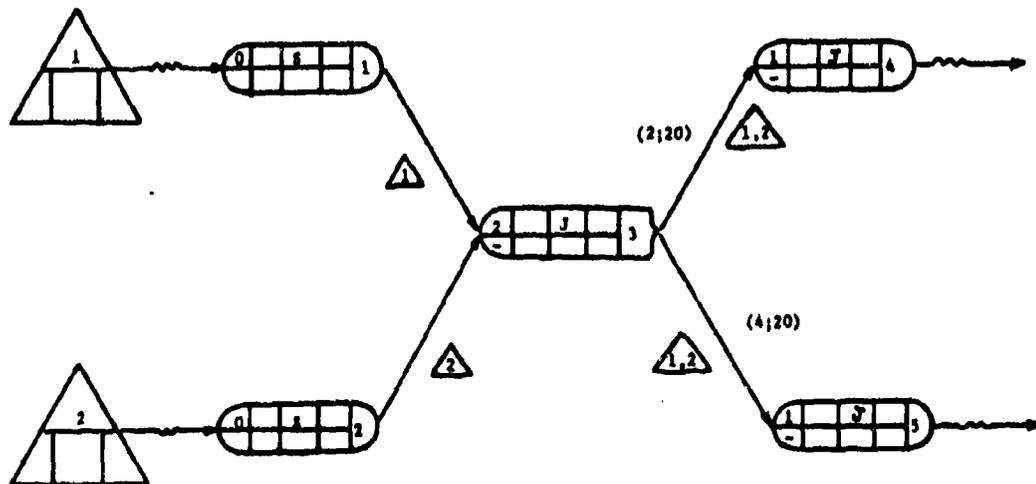


Figure 6. Parallel Tasks Leading to a Joint Task Involving Two Operators

The network in Figure 7 illustrates the situation where two operators start on the same task, and then work on separate tasks. Task 10 is a joint task and since both operators are initially set at task 10, task 10 requires operators 1 and 2. Branching from task 10 is deterministic and causes the number of requirements for tasks 11 and 12 to be reduced by 1. Since only one requirement exists for each task, the tasks are released. Operator 1 is sent to task 11 and operator 2 is sent to task 12 as indicated by the numbers in the triangles below the branches.

Figure 8 presents a network involving two operators who at first are performing tasks in parallel. The first operator that completes his task performs task 3. Since task 3 is an either task, either one of the operators may perform it. If task 3 is being performed by operator 1 when operator 2 arrives to perform the task, operator 2 goes directly to task 5 and would start it. If operator 2 arrives after task 3 was completed by operator 1, operator 2 performs task 3 over again. (If this was not desired, the network modification feature of SAINT should be used; then task 3 should be modified based on the completion of task 3.) An analogous situation occurs if operator 2 arrives at task 3 first. Branching from task 3 is deterministic, and operator 1 is sent to task 4 when he completes task 3 and operator 2 is sent to task 5 when he completes task 3.

The next example shown in Figure 9 illustrates the modification of a network based on the task completion sequence. Since only one operator is involved, there is no need to indicate the operator number beneath the branches of the network. Branching from task 2 is probabilistic and, therefore, either task 3 or 4 is released following the completion of task 2, according to the probabilities shown. If task 3 is performed, then task 5 is released (only one requirement exists) and task 5 would be started. If task 4 is performed, then task 6 replaces task 5 in the network, and task 6 is released and subsequently started.

The last example, shown in Figure 10, illustrates the clearing of operators. Two operators are involved in the mission, one starting at task 20 and the other starting at task 21. As soon as one of the operators finishes his task, it is desired to start task 22, a joint task requiring both operators. This is accomplished by clearing operator 2 from task 21 when task 20 is completed prior to task 21, and signaling task 22. This signal, plus the branch from task 20 to task 22 then causes task 22 to be released. The clearing of operator 2 makes him available to perform task 22. The completion of task 20 by operator 1 makes operator 1 available for performing task

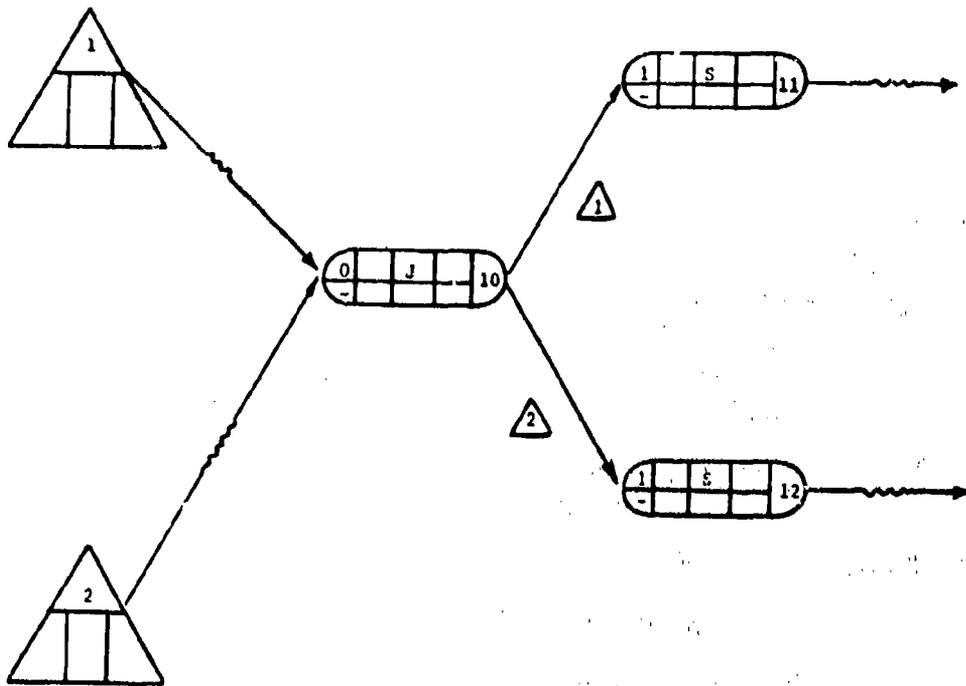


Figure 7. Joint Task Followed by Two Parallel Tasks Involving Two Operators

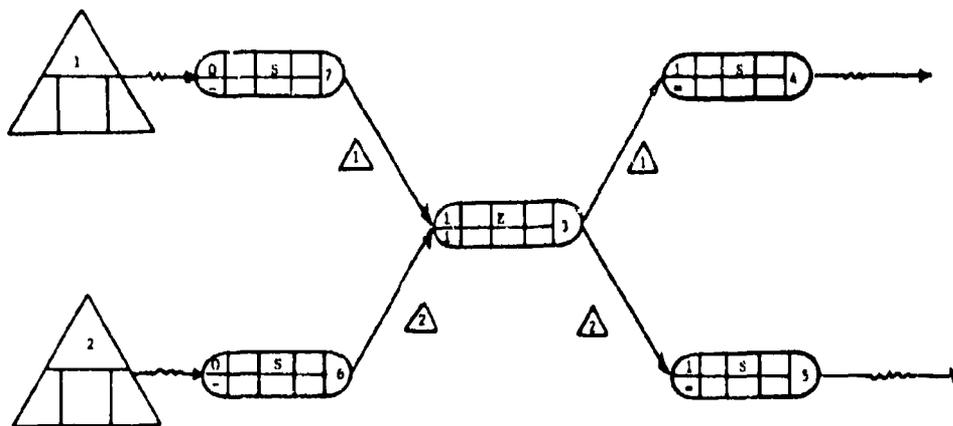


Figure 8. Parallel Tasks for Two Operators Separated by an Either Task

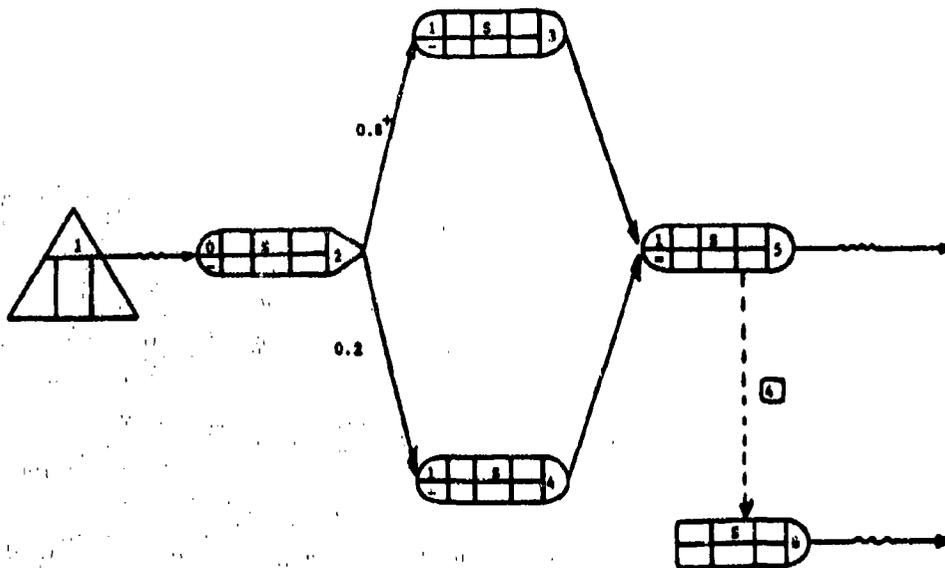


Figure 9. Network Modification Based on Task Completion Sequence

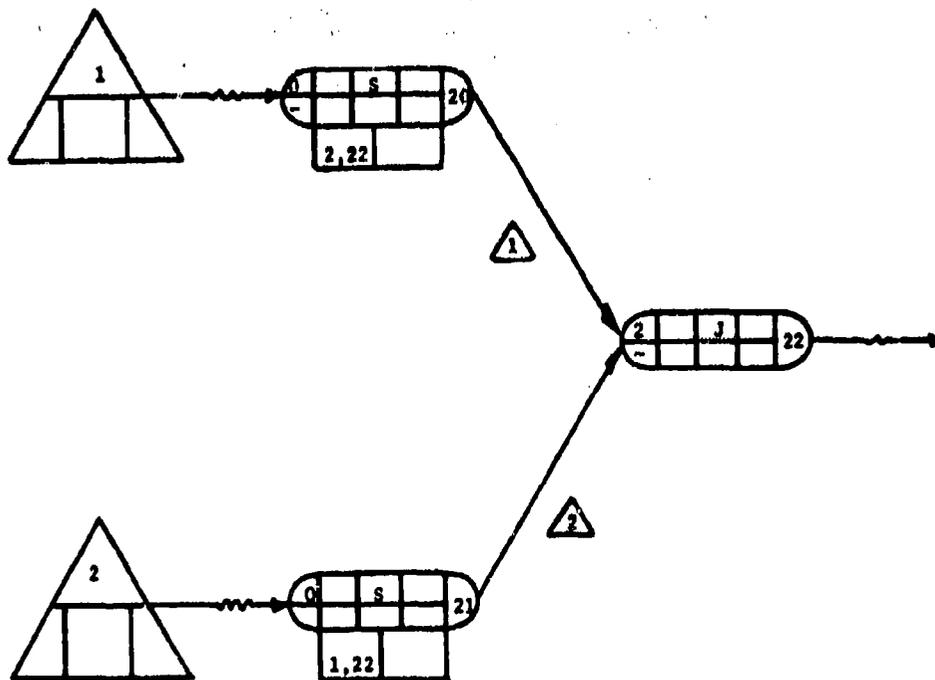


Figure 10. Clearing of Operators

22 and, hence, task 22 is started. If task 21 was completed prior to task 20, the opposite situation pertains and the clearing of operator 1 is performed and a signal to task 22 is sent.

The above examples illustrate the flexibility in modeling available from the SAINT symbol set. Not all capabilities were shown in the above examples, but hopefully the flavor of the modeling procedure was presented. The analysis obtained by the SAINT computer program is presented in the SAINT User's Manual (15) where the analysis of a large example network is presented in detail.

SECTION V

SUMMARY

A general framework has been presented for modeling and analyzing crew performance. The modeling vehicle developed is a set of network symbols and terminology and the analysis technique is simulation. The combined network modeling and simulation technique is called SAINT.

In the development of SAINT, a systems approach involving a top down analysis was employed. The type of system to be modeled was defined in terms of missions. The purpose of the modeling was established as obtaining mission performance measures. The use of SAINT allows systems designers a new flexible methodology for establishing man/machine design trade-offs throughout various stages of system development.

A mission consists of a set of tasks performed by a crew of operators having a complement of equipment in the face of environmental factors. Also associated with the mission is a scenario which defines the phases of the mission and the times by which they must be completed. Performance for the mission was defined in terms of the time required to accomplish the mission and whether the mission was accomplished successfully or resulted in a failure. These performance measures are associated with the terminal or sink tasks of the network model of the mission.

Given the above specification, network concepts were developed by which tasks are modeled as nodes and a mission is modeled as a set of tasks linked through precedence relations represented by the branches of a network. Operators flow through the network performing those tasks which are assigned to them and, in so doing, consume time. Independent variables that affect the operators performing the tasks are included in the model through parameters and functional relationships. Resource requirements are included as structural aspects of the model. Thus, SAINT can be used to model and analyze diverse missions involving operators performing tasks.

From the user's standpoint, SAINT represents a bottom-up approach. The designer or system engineer as a user must define the tasks of a mission and the relationships that exist between tasks. In defining the tasks, he is working at the elemental level which permits the obtaining of data about specific tasks. When collecting data, the user need not be concerned about total mission performance as SAINT is designed to do this. Thus, SAINT allows the user to concentrate on specific details of a task and from such information SAINT integrates the data to obtain mission performance estimates.

The decomposition of an overall problem into its elements and then providing a vehicle to integrate the elements into system performance measures is in the true spirit of the systems approach. SAINT provides this capability. As with any general technique, care and experience are necessary. Presumably we have the caution to proceed slowly and, hopefully, the courage to gain experience.

The primary recommendation resulting from this research is that applications of SAINT need to be made. Through application, modifications, deletions and extensions are anticipated. SAINT has been designed in a modular form to facilitate such possibilities.

With regard to specific areas of continuing development, the following are proposed: 1) verification of the factors and relations included in the characterization of task performance; 2) development of new concepts in order to model tasks that require continuous monitoring, queueing, and resource allocation; 3) extend the treatment of task type and the method by which operators are assigned to tasks; 4) extend SAINT to include other system and mission performance measures such as reliability, availability, and cost effectiveness.

REFERENCES

1. Adams, J. A. and C. E. Weber, "Monte Carlo Model of Tracking Performance," Human Factors, 5, 1, Feb. 1963, 81-102.
2. Chubb, G. P., "The Use of Monte Carlo Simulation to Reflect the Impact Human Factors Can Have on Systems Performance," proceedings of the 1971 Winter Simulation Conference (Fifth Conference on the Applications of Simulation), held 8-10 December 1971, New York, sponsored by ACM/ AIIE/ IEEE/SHARE/SCI/TIMS, also identified as AMRL-TR-71-75.
3. Chubb, G. P., "Using Monte Carlo Simulation to Assess the Impact Radiation Induced Performance Degradation Can Have on Mission Success," presented at the Navy Symposium on Computer Simulation as Related to Manpower and Personnel Planning, Naval Personnel Research and Development Laboratories, Washington, D.C., July, 1971.
4. Kurke, M. I., "Operational Sequence Diagrams in System Design," Human Factors, 3, 1, March 1961, 66-73.
5. Mitchell, M. B., R. L. Smith and R. A. Westland, Techniques for Establishing Personnel Performance Standards (TEPPS). Procedural Guide (2nd ed.), Dunlap and Associates, Inc., Santa Monica, California, 1968.
6. Pritsker, A. A. B., GERT: Graphical Evaluation and Review Technique, the RAND Corporation RM-4973-NASA, (1966).
7. Pritsker, A. A. B., Precedence GERT, Purdue University, School of Industrial Engineering Research Memorandum No. 71-14, December, 1971.
8. Pritsker, A. A. B. and W. W. Happ, "GERT: Part I - Fundamentals" J. Ind. Eng., Vol. XVII, No. 5, p. 267 (1966).
9. Pritsker, A. A. B. and P. J. Kiviat, GASP II, A FORTRAN-Based Simulation Language, Prentice-Hall, Inc., Englewood Cliffs, N. J. (1969).

REFERENCES (Cont.)

10. Seifert, D. J., and G. P. Chubb, Computer Models of Man-Machine Survivability/Vulnerability, AMRL-TR-72-69, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, April 1973. AD 762 528.
11. Siegel, A. I., and J. J. Wolf, Man-Machine Simulation Models, John Wiley and Sons, Inc., New York, 1969.
12. Siegel, A. I., J. J. Wolf, M. A. Fischl, W. Miehle and G. P. Chubb, Modification of the Siegel-Wolf Operator Simulation Model for On-Line Experimentation, AMRL-TR-71-60, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, June, 1971.
13. Siegel, A. I., J. J. Wolf, and R. T. Sorenson, Techniques for Evaluating Operator Loading in Man Machine Systems: Evaluation of a One or a Two Operator Evaluative Model Through a Controlled Laboratory Test, Applied Psychological Services, Inc., Wayne, Pennsylvania, 1962.
14. Whitehouse, G. E., Systems Analysis and Design Using Network Techniques, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1973.
15. Wortman, D. B., A. A. B. Pritsker, C. Seum and D. J. Seifert, SAINT: Volume II. User's Manual, AMRL-TR-73-128, Aerospace Medical Laboratory, Wright-Patterson Air Force Base, Ohio.
16. Wulff, J. J., J. N. Leonard and A. F. Pixley, A Descriptive Model for Determining Optimal Human Performance in Systems, Volume I, National Aeronautics and Space Administration, Washington, D.C., January, 1968.